

サーバ管理演習における誤り原因特定のためのコマンド実行履歴とファイル編集履歴の可視化手法の検討

西村 一輝¹ 井垣 宏¹

概要: クラウドコンピューティングや IoT などの発展により、サーバ管理技術を有するエンジニアの育成が重要になっている。そのため、大学などの教育機関において AWS などの仮想サーバ環境を用いたサーバ管理演習が行われている。サーバ管理演習とは、サーバを構築管理するための技術やノウハウを学べる実習型の演習である。そこで我々は先行研究において、学生の進捗状況把握や課題の誤りを特定する事を目的として、サーバ管理演習におけるログを 3 種類の履歴として記録するログ収集環境と収集した履歴の一つであるサーバ状況確認履歴の可視化環境の提案を行った。本稿では、収集されているコマンド実行履歴とファイル編集履歴の可視化を行う事により、サーバ状況確認履歴から学生の課題の誤りを特定できるだけでなく、学生が演習課題に行き詰まった場合の原因究明が行える可視化環境の提案を行う。

1. はじめに

クラウドコンピューティング技術の発展と普及に伴い、大学などの教育機関においてもサーバ管理技術を学ぶ演習が数多く実施されるようになりつつある。例えば、文部科学省の教育プロジェクトである enpit[6] でも、クラウド技術を対象とした PBL 型実践教育やセキュリティ技術者育成のための授業において、実際にサーバの立ち上げや設定を行うサーバ管理演習が開講されている。これらの演習では、Amazon Web Services^{*1}のようなパブリッククラウドと言われるクラウドサービスを利用することで、学生一人ひとりがそれぞれ自身のサーバを持ち、サーバソフトウェアのインストールや設定を伴う演習に取り組むような、非常に実践的な取り組みが行われている [5]。

一方で、これらの仮想マシンを利用したサーバ管理演習では、教員が全学生の進捗状況を把握することが困難という課題がある。学生が演習に取り組む際、各々のサーバに関するスキルの差やタイピングスピードによって演習を進めるスピードが異なる。また、コマンドをどの順序で実行するか、設定ファイルをどのように書き換えるかなども異なるため、どの学生がどの課題を解いたか、あるいは解けていないかを教員による目視や口頭での確認のみで把握することは困難である。

また、学生がコマンドや設定ファイルの誤記等で誤った

際に、どのような経緯でそうなったかを後から正確に把握することは難しい。学生は、課題に行き詰まると教員に質問を行う。教員は質問を受け、その学生のサーバの状況がどうなっているのかを調べて、どこが誤っているのかを特定する。この過程が簡単な演習である場合は教員にかかる負担は少ないが高度な演習になればなるほど、誤っている箇所を特定することは困難になる。また、誤っている箇所の特定だけでなく、何が原因でそのような誤りが発生したかという情報は学生へのフィードバックという観点では非常に重要であるが、原因の究明が難しい場合も多い。

これらの問題を解決する為に、我々は先行研究においてサーバ管理演習における 3 種類のログを収集するログ収集環境の提案を行った [3][2][1]。さらに、このログ収集環境を用いて、実際のサーバ管理演習によるログ収集を行い、収集したログを元にサーバ状況確認履歴可視化手法の提案を行なった。本稿では、可視化を行なったサーバ状況確認履歴に加えて、既にログ収集環境により収集していたコマンド実行履歴とファイル編集履歴の可視化手法について検討する。サーバ状況確認履歴では、教員に過度の負荷をかけることなく学生の進捗状況を一覧化することが可能となった。コマンド実行履歴とファイル編集履歴の可視化では、学生がどのような経緯によりサーバの設定を誤ったのか、また、それらの原因は何であるのかを特定する、といったより詳細な原因の究明が可能となると考えている。

¹ 大阪工業大学
Osaka Institute of Technology, 1-79-1 Kitayama, Hirakata
City, Osaka, 573-0196 Japan

^{*1} <https://aws.amazon.com/jp/>

2. サーバ管理演習とその課題

2.1 サーバ管理演習とは

サーバ管理演習は物理マシンあるいは仮想マシンを対象として、学生が様々なサーバソフトウェアのインストールや設定を実際に行うことでサーバ管理技術を学ぶ形式の演習である。こうした演習は様々なところで実施されているおり、enpit プロジェクトのクラウドコンピューティング分野である Cloud Spiral[6]では、Amazon Web Service 等のサービスを利用して仮想化されたサーバ環境のもとで演習が行われている。サーバ環境は Linux OS を対象としたものが多いが、授業によっては Windows Server や特殊な組み込み系の OS などを対象とすることもある。本研究では、Linux OS を対象としたサーバ管理演習を想定している。最近では、仮想化技術を対象とした演習や Web UI を利用したサーバ管理演習のための支援環境 [4][9][7][8] など、サーバ管理を対象とした教育やその支援手法についての研究が複数実施されている。

2.2 サーバ管理演習の流れ

学生らは演習開始時に、サーバ環境に ssh クライアントを利用してアクセスを行い、教員が事前に用意したテキストやドキュメントを元に演習を行なっていく。教員は、演習の段階ごとにサーバの状況が適切であるかを確認し、各学生が正しく演習を進められているのかを定期的に確認する。学生が正しく演習を進められている場合、段階ごとの確認手段を達成していくことで先の課題を行うことができる。学生がどこかの課題で行き詰まっている場合、学生は何が原因でサーバが不具合を起こしているのかを特定し、必要に応じて教員にサポートを依頼し、修正しなければならない。

2.3 サーバ管理演習の課題

演習は学生単位で行うため、既に演習内容に対する知識を保有している場合や、CUI などの操作に慣れている、タイピングスピードの違いなどにより、演習を進めるスピードが変わってしまう。そのため、学生毎の進捗状況を教員が把握するためには、学生に口頭で確認するだけでなく、サーバがどのような状態にあるかを定期的に確認するための手段が必要になる。

また、サーバ設定による不具合の原因は、ソフトのインストール漏れや誤ったコマンドの実行、サーバの設定ミスなど多岐にわたる。そのため、不具合の原因特定は基本的に学生自身が行うべきであるが、教員やサポートスタッフなどのサポートが求められることも少なくない。教員は学生のサーバ環境で不具合が発生した場合、現状のサーバの状況と、学生が行なった操作によって、サーバの状態

がどのように変更されたのかを確認する。これらは、ログファイルやインストールされているソフトの設定内容を確認することで知ることができるが、学生がいつ、どこで、どのような操作をして現在のサーバの状態に至ったのかを後から確認することは、学生自身の記憶やサーバに保存されるログだけでは、正確に把握することが難しい。また、学生がどのような原因で不具合を起こすのかは、今後の演習内容の改善にも繋がる為、非常に重要である。

そこで、我々は先行研究において [3][2][1] 学生の操作履歴をログとして記録するログ収集環境を提案した。以降では、3 節において我々が先行研究において実施した学生の様々な操作履歴の収集やサーバ状況確認履歴の概要について述べる。その後、4 節では本稿で提案するコマンド実行履歴、ファイル編集履歴の可視化手法の詳細について詳述する。

3. サーバ管理演習におけるコマンド実行履歴、ファイル編集履歴及びサーバ状況確認履歴の収集

サーバ管理演習において、学生はサーバに対して何らかのコマンドの実行やファイルの編集を行い、課題に沿った形に変更していく。例えば、Web サーバを動作させる場合、yum などのパッケージ管理コマンドを用いて Web サーバのインストールを行い、設定ファイルに編集を加え、サーバを起動する。さらに、外部からサーバのテストページにアクセスできるのかを確認することで、Web サーバが正しく動作しているか確認する。

我々は先行研究 [2] において、こうした学生による一連の操作をログとして記録し、振り返りを行えるようにコマンド実行履歴、ファイル編集履歴、サーバ状況確認履歴の 3 種類のログを収集した。以降ではこの 3 種類のログの概要を述べる。

3.1 コマンド実行履歴

コマンド実行履歴とは、いつどの学生がどのようなコマンドを実行したのか、その結果が何であったかを記録したものである。収集される項目を以下に示す。

- id: 学生番号
- date: 実行時刻
- cmd: 実行したコマンド
- res: 実行した結果

ここでは Linux 環境でよく利用される history と script*2 の 2 種類の機能を利用している。

3.2 ファイル編集履歴

サーバ管理演習では、学生がインストールしたサーバソ

*2 <https://linuxjm.osdn.jp/html/util-linux/man1/script.1.html>

ソフトウェアを演習の意図通りに動作させるために、様々な設定ファイルを編集する必要がある。そこでファイル編集履歴として、誰が、いつ、どのファイルをどのように変更したか、を版管理システムの git^{*3}を利用して記録する。なお、どのファイルがいつ変更されたかを取得するために、inotify^{*4}という機能を利用している。inotify は特定のディレクトリやファイルを監視し、ファイルシステムイベントを検知するコマンドである。ファイルの変更が inotify によって検知されると、そのファイルは、予め用意されている /git ディレクトリにファイルがコピーされ、同時にそのディレクトリ内にあるローカルリポジトリ上にコミットされるという仕組みになっている。

3.3 サーバ状況確認履歴

我々が想定するサーバ管理演習において、教員が事前に決めた段階ごとに演習が進められるのであれば、何らかのコマンドを実行することによりサーバの現状を知ることが可能である。例えば、ソフトウェアが正しくインストールできているかやサーバが動作しているか、外部からのアクセスに対して特定の返答を返すかといった形式で確認できる。そのため、確認用のスクリプトやコマンドを実行する環境を用意し、どのようなスクリプトがいつ、どの学生の環境に対して実行され、結果がどうであったかをログとして記録している。

履歴として保存される情報は以下に示すとおりである。

- id: 学生番号
- date: 実行時刻
- script: 課題に対応したシェルスクリプト
- res: 実行結果

先行研究では学生の進捗状況把握のために、このサーバ状況確認履歴の可視化を行った。図 1 に Web アプリとして構築されたサーバ状況確認履歴表示画面を提示する。

この画面では、時刻の項目にあるスライダーを選択することでサーバ状況確認コマンドが実行された時刻を、サーバ状況確認コマンド項目にあるドロップダウンリストを選択することで、実際に確認したコマンドをそれぞれ選択することができる。

図 1 は実際に時刻として 11 時 10 分を、サーバ状況確認コマンドとして、Step4 の `sudo service tomcat8 status` を選択し、send ボタンをクリックしたときの画面を示している。この画面では、学生番号、実行時間、標準出力、標準エラー出力がサーバ状況確認コマンドの実行結果を示している。ソーティングが可能な表として学生番号及び、実行結果が表示されるため、標準出力やエラー出力が他の学生と異なっている学生なども容易に確認することができる。

*3 <https://git-scm.com/>

*4 https://linuxjm.osdn.jp/html/LDP_man-pages/man7/inotify.7.html

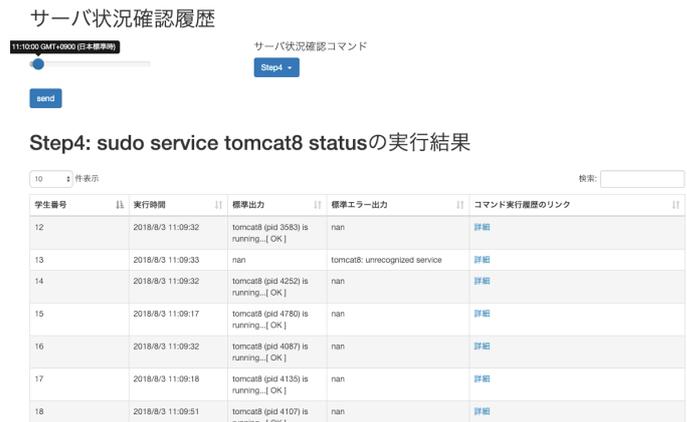


図 1 サーバ状況確認履歴表示画面

実際に図 1 では、学生番号が 13 のサーバでまだ tomcat8 がインストールされていないことが確認できる。

4. コマンド実行履歴とファイル編集履歴の可視化

サーバ状況確認履歴を可視化することにより、サーバ管理演習における課題の一つである学生の進捗状況を可視化することが可能となった。しかし、サーバ状況確認履歴だけではサーバが課題で意図されている振る舞いをしなくなった際に、その原因が何であるかや学生が行き詰まるまでの過程を知ることは困難である。そこで本研究では、学生がいつ何をどのようにした結果、サーバが正しい振る舞いをしなくなったのかをブラウザ上で教員や学生が容易に確認できるよう、残り 2 つの履歴の可視化を目指す。以降ではコマンド実行履歴とファイル編集履歴、それぞれの可視化画面について述べた後、それらを実際のログで参照した際にどのような結果が得られるかを詳述する。

4.1 コマンド実行履歴表示画面

図 2 に示したサーバ状況確認履歴表示画面では、マトリクスの一右側に「詳細」と書かれたリンクがある。

Step2: dateの実行結果

学生番号	実行時間	標準出力	標準エラー出力	コマンド実行履歴のリンク
0	2018/8/3 10:55:48	Fri Aug 3 01:55:48 UTC 2018	nan	詳細
1	2018/8/3 10:55:48	Fri Aug 3 10:55:48 JST 2018	nan	詳細
2	2018/8/3 10:55:48	Fri Aug 3 10:55:48 JST 2018	nan	詳細
3	2018/8/3 10:55:48	Fri Aug 3 10:55:48 JST 2018	nan	詳細
4	2018/8/3 10:55:48	Fri Aug 3 10:55:48 JST 2018	nan	詳細
5	2018/8/3 10:55:48	Fri Aug 3 01:55:48 UTC 2018	nan	詳細
7	2018/8/3 10:49:03	Fri Aug 3 01:49:03 UTC 2018	nan	詳細
8	2018/8/3 10:55:48	Fri Aug 3 10:55:48 JST 2018	nan	詳細
9	2018/8/3 10:56:06	Fri Aug 3 10:56:06 JST 2018	nan	詳細
10	2018/8/3 10:56:06	Fri Aug 3 10:56:06 JST 2018	nan	詳細
学生番号	実行時間	標準出力	標準エラー出力	コマンド実行履歴のリンク

図 2 11 時に Step2 のサーバ状況確認履歴表示画面

詳細リンクを押すと、同じ行に記述された学生の指定時

刻より前に実行されたコマンドとその結果が図3で示すように一覧で表示される。実行したコマンドとその実行結果はコマンド実行履歴として記録された `bash_history` のログと `script` コマンドのログを組み合わせて表示されている。これにより、学生がいつどのようなコマンドを実行した結果、どのような結果になっているのかを可視化することが可能となる。

実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク
2018/8/3 10:55:15	date	Fri Aug 3 10:55:15 JST 2018	
2018/8/3 10:54:07	sudo reboot	(09wpts/1) at 10:54 ... The system is going down for reboot NOW!	詳細
2018/8/3 10:52:38	date	Fri Aug 3 10:52:38 JST 2018	詳細
2018/8/3 10:52:35	sudo ln -sf /usr/share/zoneinfo/Asia/Tokyo /etc/localtime		詳細
2018/8/3 10:52:04	cat /etc/sysconfig/clock	ZONE="Asia/Tokyo" UTC=true	詳細
2018/8/3 10:49:44	sudo vi /etc/sysconfig/clock	[7:049#(1~)(200#)(1:38#)(12:25#)(12#)? 28(27#min)(#)(2:47:25)(#)1# /etc/sysconfig/clock" ZL_20C(1:1#)ZONE="UTC" UTC=true	詳細
2018/8/3 10:48:20	ls	user.dat	
2018/8/3 10:47:02	date	Fri Aug 3 01:47:02 UTC 2018	
2018/8/3 10:40:34	cat user.dat	team006-3	
2018/8/3 10:40:31	echo team006-3 > user.dat		詳細
実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク

図3 11時までに実行された学生番号3のコマンド実行履歴表示画面

図3は学生番号3の学生が11時00分以前に実行したコマンドの一覧と実行結果となっている。このように実行したコマンドと結果が時系列で表示される。初期の表示画面では最新のものが一番上に来るように表示されており、ソート機能を利用して逆順にすることも可能になっている。検索機能も実装されており、特定のコマンドを実行しているか、タイプミスが発生しているかなどをキーワードを入力することで検索が可能となっている。例えば図4では、“command not found”で検索し、該当する実行結果のみを表示した例である。この例の場合、`date` コマンドのスペルを間違えた際のコマンド実行履歴が確認できている。

実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク
2018/8/3 10:51:13	data	bash: data: command not found	
実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク

図4 検索機能を利用したコマンド実行履歴表示画面

同様に、学生が設定ファイルをいつどのように編集を行ったのか、をブラウザから把握できるようにするため、ファイル編集履歴の可視化を行った。

4.2 ファイル編集履歴表示画面

ファイル編集履歴はコマンド実行履歴表示画面から確認できる。図3に示すとおり、コマンド実行履歴表示画面の最も右の列にファイル編集履歴確認画面へのリンクが表示されている。このリンクは1つ1つがファイル編集履歴として保存されている `git` リポジトリのコミットに対応している。コマンド実行履歴が時系列順にソートされているとする。そのとき、ある行のコマンド実行時刻と次の行のコマンド実行時刻の間に記録されたコミットがある場合、最初の行の末尾にそのコミットへのリンクが表示される。図3の場合、10時49分44秒に `sudo vi /etc/sysconfig/clock` が実行されている。その行の末尾に表示されているコミットは `/etc/sysconfig/clock` ファイルが10時51分49秒に編集されたことを記録したものとなっている。対象のボタンをクリックすると、そのファイルのコミットに関するページが図5のように表示される。ファイル編集履歴表示画面は `git` のリポジトリ管理サーバである `gitbucket`^{*5} を利用している。

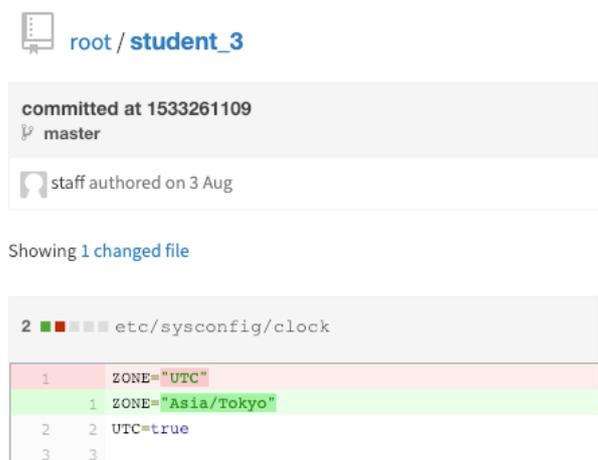


図5 ファイル編集履歴表示画面

ファイル編集履歴として保存されている `git` リポジトリはすべて `gitbucket` に `push` されており、どの学生がいつ何のファイルをどのように編集したのかを確認することができる。図5の場合、`ファイル/etc/sysconfig/clock` が `ZONE="UTC"` から `ZONE="Asia/Tokyo"` へと変更されていることが見て取れる。

次章では、実際のサーバ管理演習で収集を行った3種類の履歴を利用し、課題において何らかの誤りが発生した場合、履歴情報から学生の誤りの原因特定が行えるのかを詳述する。

*5 <https://bitbucket.org/>

5. 可視化手法のサーバ管理演習への適用

5.1 ログ収集対象の授業と教員による準備内容

enpit プロジェクトのクラウドコンピューティング分野である Cloud Spiral で行われたサーバ管理演習において、実際に我々のログ収集環境を導入し、コマンド実行履歴、ファイル編集履歴、サーバ状況確認履歴を収集した [2]。授業内容としては、仮想マシンの作成、Web サーバの導入・設定、Web アプリケーションの開発、アプリの自動ビルド・デプロイの設定や Web アプリの SSL 化などの Web アプリケーションを 1 から作成するために必要な技術を一通り学ぶことができる演習となっている。演習時間としては、朝の 10 時から 18 時半までの 1 日行い、授業規模としては大学院生の計 43 人が授業に参加している。

学生らは Amazon Web Services を利用して本演習を実施した。そのため、教員はまず My AMI(Amazon Machine Image) としてログ収集環境を導入したテンプレートを作成した。さらに教員は、サーバ状況確認コマンドとして 25 個のコマンドを事前に用意した。これらの 25 個のコマンドは 5 分に 1 回学生が作成したすべての仮想マシンを対象として、ログを収集するために用意した仮想マシンから実行された。実行結果について、ファイル編集履歴は git リポジトリとして、コマンド実行履歴とサーバ状況確認履歴は mongodb*6 に保存された。なお、本演習で収集したサーバ状況確認履歴は 90595 件、全学生が演習中に実行したコマンドの実行数は 4679 回となった。

5.2 3 種類の可視化により得られた演習中の誤り事例

ここからは収集したログを用いて、サーバ状況確認履歴、コマンド実行履歴、ファイル編集履歴を可視化することにより、学生が課題に対して行き詰まったタイミングとその原因の究明を行えるか調査を行なった。

11 時 40 分に Step8 の課題に対してサーバ状況確認履歴を参照し、標準出力の列にソートをかけると現時点においてまだ学生番号 20, 39, 40 の 3 人の学生が課題をクリア出来ていないことが図 6 から読み取れる。以降では、この 3 人の誤り事例から原因の究明を行う。Step8 の課題内容は、公式ドキュメントを参考に mongodb のインストールを行うことであり、インストールの手順として、リポジトリの登録を行うためのファイルを作成し、その後 yum コマンドを用いてインストールを行うというものになっている。

まず、図 6 から学生番号 20 番の学生が課題をクリア出来ていないことがわかる。そこで、コマンド実行履歴のリンクから図 7 が示すコマンド実行履歴表示画面を参照する。

図 7 から mongodb をインストールするために yum コマンドを実行しているが、正常にインストールできていないこ

Step8: sudo service mongod statusの実行結果

学生番号	実行時間	標準出力	標準エラー出力	コマンド実行履歴のリンク
20	2018/8/3 11:39:07	nan	mongod: unrecognized service	詳細
39	2018/8/3 11:39:14	nan	mongod: unrecognized service	詳細
40	2018/8/3 11:39:15	nan	mongod: unrecognized service	詳細
32	2018/8/3 11:39:11	mongod is stopped	nan	詳細
41	2018/8/3 11:39:14	mongod is stopped	nan	詳細
25	2018/8/3 11:39:07	mongod (pid 8477) is running...	nan	詳細

図 6 11 時 40 分時点のサーバ状況確認履歴表示画面

実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク
2018/8/3 11:38:48	sudo yum install -y mongodb-org-3.4.repo	Loaded plugins: priorities, update-mo... upgrade-helper amazon-ian 2.1 kB 00:00:00 amazon-updates 2.5 kB 00:00:00 mongodb-org-3.4 2.5 kB 00:00:00 mongodb-org-3.4(primary, db 0%) 0.0 B s 0 B ----- ETA mongodb-org-3.4(primary, db) 84 kB 00:00:00 No package [!mongodb-org-3.4.repo] available. Error: Nothing to do	

図 7 11 時 40 分までに実行された学生番号 20 のコマンド実行履歴表示画面

とが実行結果から確認できる。そこで、実行したコマンドを確認すると、sudo yum install -y mongodb-org と入力していることがわかる。この場合、正しくは sudo yum install -y mongodb-org であり、学生が誤ったコマンドを実行しているという原因の特定ができた。

次に、図 6 から学生番号 40 番の学生が課題をクリア出来ていないことがわかる。そこで、コマンド実行履歴のリンクから図 8 が示すコマンド実行履歴表示画面を参照する。

実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク
2018/8/3 11:39:07	sudo service mongod status	mongod: unrecognized service	
2018/8/3 11:38:50	mongo	bash: mongo: command not found	
2018/8/3 11:35:27	sudo yum install -y mongodb-org	Loaded plugins: priorities, update-mo... upgrade-helper No package [!mongodb-org] available. Error: Nothing to do	
2018/8/3 11:35:08	enabled=1		
2018/8/3 11:35:09	[mongodb-org-2.6]	bash: [mongodb-org-2.6]: command not found bash: 2.6: command not found /usr/x86_64/	
2018/8/3 11:35:06	name=MongoDB 2.6 Repository	bash: [mongodb-org-2.6]: command not found bash: 2.6: command not found /usr/x86_64/	
2018/8/3 11:35:09	baseurl=http://download.mongodb.org/linux/updates/x86_64/	bash: [mongodb-org-2.6]: command not found bash: 2.6: command not found /usr/x86_64/	
2018/8/3 11:35:06	gpgcheck=0	bash: [mongodb-org-2.6]: command not found bash: 2.6: command not found /usr/x86_64/	

図 8 11 時 40 分までに実行された学生番号 40 のコマンド実行履歴表示画面

一行目の実行結果から、学生は sudo service mongod status と実行しているが mongod: unrecognized service と表示されており、mongodb が正しくインストールされていないことが確認できる。また、ログを遡ると”command not found”と表示されている部分が見受けられる。この Step では本来ならば、mongodb のリポジトリ登録用のファイルを指定のディレクトリに対して作成してから、yum コマンドを実行しなければならない。しかしながら、この学生

*6 <https://www.mongodb.com/>

は誤ってファイルに記述すべき内容をコマンドとして直接実行してしまっており、リポジトリの登録が正しく行えていないため mongoddb が正しくインストール出来ないことが特定できる。

最後の事例として、図 6 から学生番号 39 番の学生が課題をクリア出来ていないことがわかる。そこで、コマンド実行履歴のリンクから図 9 が示すコマンド実行履歴表示画面を参照する。

実行時刻	実行したコマンド	実行結果	ファイル編集履歴のリンク
2018/8/3 11:36:36	cat mongoddb-org-3.4.repo	cat: mongoddb-org-3.4.repo: No such file or directory	
2018/8/3 11:36:47	sudo yum install -y mongoddb-org	Loaded plugins: priorities, update-motd, upgrade-helper File contains no section headers. file:///etc/yum.repos.d/mongoddb-org-3.4.repo, line: 1 *name=MongoDB Repository* ^	
2018/8/3 11:35:52	sudo vi /etc/yum.repos.d/mongoddb-org-3.4.repo		詳細
2018/8/3 11:33:56	sudo vi /etc/yum.repos.d/mongoddb-org-3.4		

図 9 11 時 40 分までに実行された学生番号 39 のコマンド実行履歴表示画面

本課題では、mongoddb のインストールを行うために、mongoddb のリポジトリファイルを作成している。コマンド実行履歴表示画面でも、vi コマンドを利用して学生がそのファイルを編集していることが確認できる。さらに、入力を行なったコマンドに誤りはなく、課題に対して正しいコマンドを実行しているにも関わらず、mongoddb がインストールされていないことも確認できる。そこでコマンド実行履歴表示画面の右側にあるファイル編集履歴のリンク項目の詳細をクリックし、ファイル編集履歴表示画面を参照する。以下に図 10 のファイル編集履歴表示画面を示す。

図 10 学生番号 39 のファイル編集履歴表示画面

ファイル編集履歴表示画面を参照すると、11 時 36 分 14 秒に/etc/yum.repos.d/mongoddb-org-3.4.repo ファイルを作成し、編集していることが分かる。編集内容を確認すると本来であれば一行目で記述されているべき”[mongoddb-org-3.4]”の記述が抜けていることが分かるため、yum コマ

ンドが正しく実行されないという原因が特定できた。

以上のように、サーバ状況確認履歴、コマンド実行履歴、ファイル編集履歴を組み合わせることで可視化を行うことにより、実際に行われたサーバ管理演習で発生する様々な不具合の原因特定が可能になった。

6. おわりに

サーバ状況確認履歴、コマンド実行履歴、ファイル編集履歴を組み合わせることで可視化を行った。サーバ状況確認履歴とコマンド実行履歴の 2 種類を組み合わせることで、コマンドの実行し忘れや、タイプミスなどをサーバ状況とあわせて容易に確認することができるようになった。また、サーバ状況確認履歴とコマンド実行履歴、ファイル編集履歴の全てのログを組み合わせることでログを遡ることにより、サーバが演習の意図通りになっていないときに、コマンドの実行ミスや学生がファイルを間違えて編集していたことなど、複数の不具合を容易に特定できるようになることが確認できた。今後はサーバの不具合と不具合原因の両方を学生間や過去事例との比較などに基づいて、自動的に推定するような仕組みを構築し、サーバ管理演習支援の枠組みをより充実させていきたい。

謝辞 本研究は JSPS 科研費 JP17K00500 の助成を受けたものである。

参考文献

- [1] 西村一輝, 井垣 宏: サーバ管理演習におけるサーバ状況確認履歴可視化手法の検討, 電子情報通信学会技術研究報告= IEICE technical report : 信学技報, Vol. 117, No. 381, pp. 155-160 (2018).
- [2] 西村一輝, 井垣 宏: サーバ管理演習のためのユーザー行動履歴収集システムの検討, 電子情報通信学会技術研究報告= IEICE technical report : 信学技報, Vol. 117, No. 381, pp. 155-160 (2018).
- [3] 西村一輝, 井垣 宏, 尾花将輝: 進捗状況の把握と振り返り支援を目的としたサーバ管理演習のためのユーザー行動履歴収集システムの検討, ソフトウェアエンジニアリングシンポジウム 2018 論文集, Vol. 2018, pp. 253-254 (2018).
- [4] 鎌田元樹, 榎田秀夫: OpenStack を利用したサーバ設定演習システムの提案, 研究報告インターネットと運用技術 (IOT), Vol. 2013, No. 3, pp. 1-6 (2013).
- [5] 佐伯幸郎, まつ本真佑, 井垣 宏, 福安直樹, 水谷泰治, 中村匡秀: ソフトウェア開発 PBL における AWS in Education 助成プログラムの活用, 日本ソフトウェア科学会大会論文集, No. 32, p. 5 (2015).
- [6] 春名修介, 楠本真二, 井上克郎: 実践的情報教育協働ネットワーク: enPiT, SEC journal, Vol. 10, No. 2, pp. 54-57 (2014).
- [7] 中崎満晶, 越智 徹, 中西通雄: OpenStack を用いた Web サーバ設定演習環境の構築, 電子情報通信学会技術研究報告= IEICE technical report : 信学技報, Vol. 115, No. 482, pp. 49-54 (2016).
- [8] 川橋 裕: 情報危機管理における演習環境の構築と運用, 電子情報通信学会技術研究報告= IEICE technical report : 信学技報, Vol. 20, No. 3, pp. 239-248 (2010).
- [9] 星野裕樹, 納富一宏, 西村広光, 示野浩士: サーバ管理者養成を目的とした学習支援システムの開発と評価, 電気学

会論文誌. C, Vol. 136, No. 7, pp. 986–994 (オンライン),
DOI: 10.1541/ieejciss.136.986 (2016).